

Quantum convolutional neural networks

Iris Cong¹, Soonwon Choi^{1,2*} and Mikhail D. Lukin¹

Neural network-based machine learning has recently proven successful for many complex applications ranging from image recognition to precision medicine. However, its direct application to problems in quantum physics is challenging due to the exponential complexity of many-body systems. Motivated by recent advances in realizing quantum information processors, we introduce and analyse a quantum circuit-based algorithm inspired by convolutional neural networks, a highly effective model in machine learning. Our quantum convolutional neural network (QCNN) uses only $O(\log(N))$ variational parameters for input sizes of N qubits, allowing for its efficient training and implementation on realistic, near-term quantum devices. To explicitly illustrate its capabilities, we show that QCNNs can accurately recognize quantum states associated with a one-dimensional symmetry-protected topological phase, with performance surpassing existing approaches. We further demonstrate that QCNNs can be used to devise a quantum error correction scheme optimized for a given, unknown error model that substantially outperforms known quantum codes of comparable complexity. The potential experimental realizations and generalizations of QCNNs are also discussed.

The complex nature of quantum many-body systems motivates the use of machine learning techniques to analyse them. Indeed, large-scale neural networks have successfully solved classically difficult problems such as image recognition or optimization of classical error correction¹, and their architectures have been related to various physical concepts^{2,3}. As such, a number of recent works have used neural networks to study properties of quantum many-body systems^{4–10}. However, the direct application of these classical algorithms is challenging for intrinsically quantum problems, which take quantum states or processes as inputs. This is because the extremely large many-body Hilbert space hinders the efficient translation of such problems into a classical framework without performing exponentially difficult quantum state or process tomography^{11,12}.

Recent experimental progress towards realizing quantum information processors^{13–16} has led to proposals for the use of quantum computers to enhance conventional machine learning tasks^{17–20}. Motivated by such developments, we introduce and analyse a machine learning-inspired quantum circuit model—the quantum convolutional neural network (QCNN)—and demonstrate its ability to solve important classes of intrinsically quantum many-body problems. The first class of problems we consider is quantum phase recognition (QPR), which asks whether a given input quantum state ρ_{in} belongs to a particular quantum phase of matter. In contrast to many existing schemes based on tensor network descriptions^{21–23}, we assume that ρ_{in} is prepared in a physical system without direct access to its classical description. The second class, quantum error correction (QEC) optimization, asks for an optimal QEC code for a given, a priori unknown error model such as dephasing or potentially correlated depolarization in realistic experimental settings. We provide both theoretical insight and numerical demonstrations for the successful application of a QCNN to these important problems, and show its feasibility for near-term experimental implementation.

QCNN circuit model

Convolutional neural networks provide a successful machine learning architecture for classification tasks such as image recognition^{1,24,25}. A CNN generally consists of a sequence of different

(interleaved) layers of image processing; in each layer, an intermediate two-dimensional array of pixels, called a feature map, is produced from the previous one (Fig. 1a). (More generally, CNN layers connect ‘volumes’ of multiple feature maps to subsequent volumes; for simplicity, we consider only a single feature map per volume and leave the generalization to future works.) The convolution layers compute new pixel values $x_{ij}^{(\ell)}$ from a linear combination of nearby ones in the preceding map $x_{ij}^{(\ell)} = \sum_{a,b=1}^w w_{a,b} x_{i+a,j+b}^{(\ell-1)}$,

where the weights $w_{a,b}$ form a $w \times w$ matrix. Pooling layers reduce feature map size, for example by taking the maximum value from a few contiguous pixels, and are often followed by the application of a nonlinear (activation) function. Once the feature map size becomes sufficiently small, the final output is computed from a function that depends on all remaining pixels (the fully connected layer). The weights and fully connected function are optimized by training on large datasets. In contrast, variables such as the number of convolution and pooling layers and the size w of the weight matrices (known as hyperparameters) are fixed for a specific CNN¹. The key properties of a CNN are thus translationally invariant convolution and pooling layers, each characterized by a constant number of parameters (independent of system size) and sequential data size reduction (that is, a hierarchical structure).

Motivated by this architecture, we introduce a QCNN circuit model extending these key properties to the quantum domain (Fig. 1b). The circuit’s input is an unknown quantum state ρ_{in} . A convolution layer applies a single quasilocal unitary (U) in a translationally invariant manner for finite depth. For pooling, a fraction of qubits are measured, and their outcomes determine unitary rotations (V_j) applied to nearby qubits. Hence, nonlinearities in QCNNs arise from reducing the number of degrees of freedom. Convolution and pooling layers are performed until the system size is sufficiently small; then, a fully connected layer is applied as a unitary F on the remaining qubits. Finally, the outcome of the circuit is obtained by measuring a fixed number of output qubits. As in the classical case, QCNN hyperparameters such as the number of convolution and pooling layers are fixed, and the unitaries themselves are learned.

¹Department of Physics, Harvard University, Cambridge, MA, USA. ²Department of Physics, University of California, Berkeley, Berkeley, CA, USA.

*e-mail: soonwon@berkeley.edu

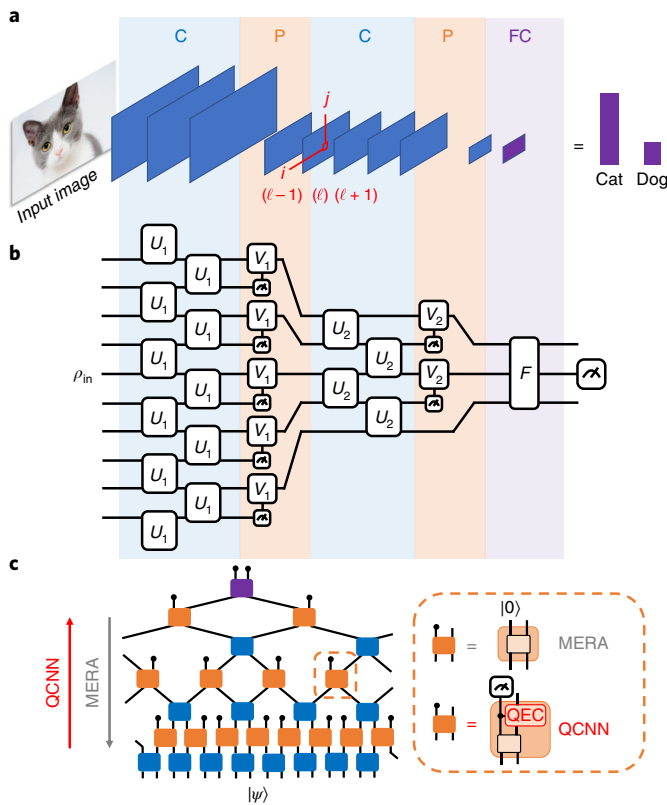


Fig. 1 | The concept of QCNNs. **a**, Simplified illustration of classical CNNs. A sequence of image-processing layers transforms an input image into a series of feature maps (blue rectangles) and finally into an output probability distribution (purple bars). C, convolution; P, pooling; FC, fully connected. **b**, QCNNs inherit a similar layered structure. Boxes represent unitary gates or measurement with feed-forwarding. **c**, The QCNN and the MERA share the same circuit structure, but run in reverse directions. Image of cat from <https://www.pexels.com/photo/grey-and-white-short-fur-cat-104827/>.

A QCNN to classify N -qubit input states is thus characterized by $O(\log(N))$ parameters. This corresponds to a double exponential reduction compared with a generic quantum circuit-based classifier¹⁹ and allows for efficient learning and implementation. For example, given a set of M classified training vectors $\{(|\psi_\alpha\rangle, y_\alpha): \alpha = 1, \dots, M\}$, where $|\psi_\alpha\rangle$ are input states and $y_\alpha = 0$ or 1 are corresponding binary classification outputs, one could compute the mean squared error

$$\text{MSE} = \frac{1}{2M} \sum_{\alpha=1}^M \left(y_i - f_{\{U_i, V_j, F\}}(|\psi_\alpha\rangle) \right)^2 \quad (1)$$

Here, $f_{\{U_i, V_j, F\}}(|\psi_\alpha\rangle)$ denotes the expected QCNN output value for input $|\psi_\alpha\rangle$. Learning then consists of initializing all unitaries and successively optimizing them until convergence, for example via gradient descent.

To gain physical insight into the mechanism underlying QCNNs and motivate their application to the problems under consideration, we now relate our circuit model to two well-known concepts in quantum information theory—the multiscale entanglement renormalization ansatz²⁶ (MERA) and QEC. The MERA framework provides an efficient tensor network representation of many classes of interesting many-body wavefunctions, including those associated with critical systems^{26–28}. A MERA can be understood as a quantum state generated by a sequence of unitary and isometry layers applied

to an input state (for example $|00\rangle$). While both types of layers apply quasilocality unitary gates, each isometry layer first introduces a set of new qubits in a predetermined state, such as $|0\rangle$ (Fig. 1c). This exponentially growing, hierarchical structure allows for the long-range correlations associated with critical systems. The QCNN circuit has similar structure but runs in the reverse direction. Hence, for any given state $|\psi\rangle$ with a MERA representation, there is always a QCNN that recognizes $|\psi\rangle$ with deterministic measurement outcomes; one such QCNN is simply the inverse of the MERA circuit.

For input states other than $|\psi\rangle$, however, such a QCNN does not generally produce deterministic measurement outcomes. These additional degrees of freedom distinguish a QCNN from a MERA. Specifically, we can identify the measurements as syndrome measurements in QEC²⁹, which determine error correction unitaries V_j to apply to the remaining qubit(s). Thus, a QCNN circuit with multiple pooling layers can be viewed as a combination of a MERA (an important variational ansatz for many-body wavefunctions) and nested QEC (a mechanism to detect and correct local quantum errors without collapsing the wavefunction). This makes QCNNs a powerful architecture for classifying input quantum states or devising new QEC codes. In particular, for QPR, the QCNN can provide a MERA realization of a representative state $|\psi_0\rangle$ in the target phase. Other input states within the same phase can be viewed as $|\psi_0\rangle$ with local errors, which are repeatedly corrected by the QCNN in multiple layers. In this sense, the QCNN circuit can mimic renormalization-group flow, a methodology that successfully classifies many families of quantum phases³⁰. For QEC optimization, the QCNN structure allows for simultaneous optimization of efficient encoding and decoding schemes with potentially rich entanglement structure.

Detecting a 1D symmetry-protected topological phase

We first demonstrate the potential of a QCNN by applying it to QPR in a class of 1D many-body systems. Specifically, we consider a $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry-protected topological (SPT) phase \mathcal{P} , a phase containing the $S=1$ Haldane chain³¹, and ground states $\{|\psi_G\rangle\}$ of a family of Hamiltonians on a spin-1/2 chain with open boundary conditions:

$$H = -J \sum_{i=1}^{N-2} Z_i X_{i+1} Z_{i+2} - h_1 \sum_{i=1}^N X_i - h_2 \sum_{i=1}^{N-1} X_i X_{i+1} \quad (2)$$

where X_i, Z_i are Pauli operators for the spin at site i , and h_1, h_2 and J are parameters of the Hamiltonian. The $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry is generated by $X_{\text{even(odd)}} = \prod_{i \in \text{even(odd)}} X_i$. Figure 2a shows the phase diagram

as a function of $(h_1/J, h_2/J)$. When $h_2=0$, the Hamiltonian is exactly solvable via the Jordan–Wigner transformation³⁰, confirming that \mathcal{P} is characterized by non-local order parameters. When $h_1=h_2=0$, all terms are mutually commuting, and a ground state is the 1D cluster state. Our goal is to identify whether a given, unknown ground state drawn from the phase diagram belongs to \mathcal{P} .

As an example, we first present an exact, analytical QCNN circuit that recognizes \mathcal{P} (Fig. 2b). The convolution layers involve controlled-phase gates as well as Toffoli gates with controls in the X -basis, and pooling layers perform phase-flips on remaining qubits when one adjacent measurement yields $X=-1$. This convolution–pooling unit is repeated d times, where d is the QCNN depth. The fully connected layer measures $Z_{i-1} X_i Z_{i+1}$ on the remaining qubits. Figure 2c shows the QCNN output for a system of $N=135$ spins and $d=1, \dots, 4$ along $h_1=0.5J$, obtained using matrix product state simulations. As d increases, the measurement outcomes show sharper changes around the critical point, and the output of a $d=2$ circuit already reproduces the phase diagram with high accuracy (Fig. 2a). This QCNN can also be used for other Hamiltonian models belonging to the same phase, such as the $S=1$ Haldane chain³¹ (see Methods).

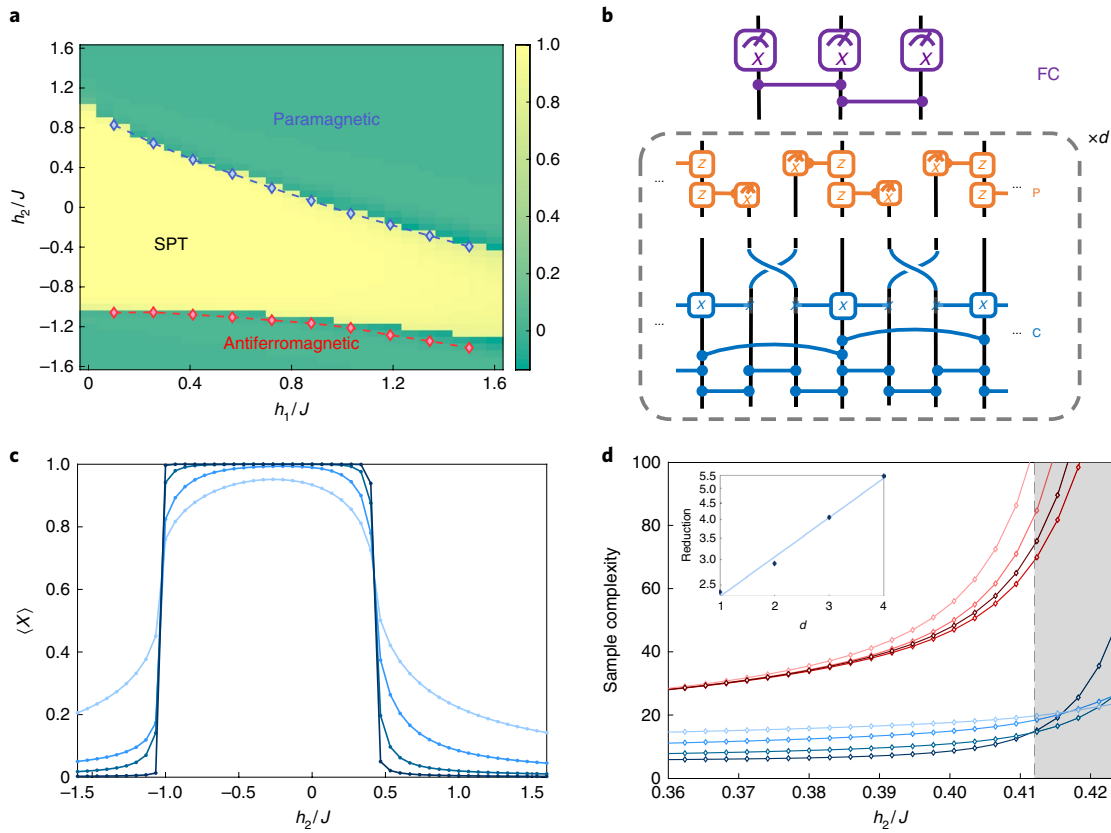


Fig. 2 | Application to quantum phase recognition. **a**, The phase diagram of the Hamiltonian in the main text. The phase boundary points (blue and red diamonds) are extracted from infinite-size DMRG numerical simulations, while the background shading (colour scale) represents the output from the exact QCNN circuit for input size $N = 45$ spins (see Methods). **b**, Exact QCNN circuit to recognize a $\mathbb{Z}_2 \times \mathbb{Z}_2$ SPT phase. Blue line segments represent controlled-phase gates, blue three-qubit gates are Toffoli gates with the control qubits in the X basis, and orange two-qubit gates flip the target qubit's phase when the X measurement yields -1 . The fully connected layer applies controlled-phase gates followed by an X_i projection, effectively measuring $Z_{i-1}X_iZ_{i+1}$. **c**, Exact QCNN output along $h_1 = 0.5J$ for $N = 135$ spins, depths $d = 1, \dots, 4$ (from light to dark blue). **d**, Sample complexity of QCNN at depths $d = 1, \dots, 4$ (from light to dark blue) versus SOPs of length $N/2, N/3, N/5$ and $N/6$ (from light to dark red) to detect the SPT/paramagnet phase transition along $h_1 = 0.5J$ for $N = 135$ spins. The critical point is identified as $h_2/J = 0.423$ using infinite-size DMRG. In the shaded area, the correlation length exceeds the system size, and finite-size effects can considerably affect our results. Inset: the ratio of SOP sample complexity to QCNN sample complexity is plotted as a function of d on a logarithmic scale for $h_2/J = 0.3918$. In the numerically accessible regime, this reduction of sample complexity scales exponentially as $1.73e^{0.28d}$ (trendline).

Sample complexity. The performance of a QPR solver can be quantified by sample complexity¹¹: what is the expected number of copies of the input state required to identify its quantum phase? We demonstrate that the sample complexity of our exact QCNN circuit is substantially better than that of conventional methods. In principle, \mathcal{P} can be detected by measuring a non-zero expectation value of string order parameters (SOPs)^{32,33} S such as

$$S_{ab} = Z_a X_{a+1} X_{a+3} \dots X_{b-3} X_{b-1} Z_b \quad (3)$$

In practice, however, the expectation values of SOP vanish near the phase boundary due to diverging correlation length³³; since quantum projection noise is maximal in this vicinity, many experimental repetitions are required to affirm a non-zero expectation value. In contrast, the QCNN output is much sharper near the phase transition, so fewer repetitions are required.

Quantitatively, given some $|\psi_{\text{in}}\rangle$ and SOP S , a projective measurement of S can be modelled as a (generalized) Bernoulli random variable, where the outcome is 1 with probability $p = (\langle \psi_{\text{in}} | S | \psi_{\text{in}} \rangle + 1) / 2$ and -1 with probability $1 - p$ (since S^2 equals the identity operator); after M binary measurements, we estimate p . $p > p_0 = 0.5$ signifies $|\psi_{\text{in}}\rangle \in \mathcal{P}$. We define the sample complexity M_{\min} as the minimum

M to test whether $p > p_0$ with 95% confidence using an arcsine variance-stabilizing transformation³⁴:

$$M_{\min} = \frac{1.96^2}{(\arcsin \sqrt{p} - \sqrt{\arcsin p_0})^2} \quad (4)$$

Similarly, the sample complexity for a QCNN can be determined by replacing $\langle \psi_{\text{in}} | S | \psi_{\text{in}} \rangle$ by the QCNN output expectation value in the expression for p .

Figure 2d shows the sample complexity for the QCNN at various depths and SOPs of different lengths. The QCNN clearly requires substantially fewer input copies throughout the parameter regime, especially near criticality. More importantly, although the SOP sample complexity scales independently of string length, the QCNN sample complexity consistently improves with increasing depth and is limited only by finite size effects in our simulations. In particular, compared with SOPs, the QCNN reduces sample complexity by a factor that scales exponentially with the depth of the QCNN in numerically accessible regimes (inset). Such scaling arises from the iterative QEC performed at each depth and is not expected from any measurements of simple (potentially nonlocal) observables. We show in the Methods that our QCNN circuit measures a multiscale

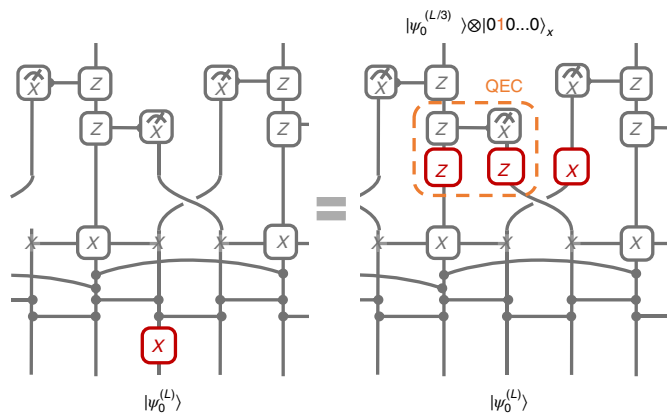


Fig. 3 | MERA and QEC in the QCNN circuit. When the input state is the cluster state with a single-qubit X error (left), the convolution–pooling unit of our circuit identifies and corrects the error, while reducing the system size by a factor of 3 (right). This process resembles the combination of MERA and QEC.

string order parameter—a sum of products of exponentially many different SOPs that remains sharp up to the phase boundary.

MERA and QEC. Further insights into the performance of the QCNN are revealed by interpreting it in terms of MERA and QEC. In particular, our QCNN is specifically designed to contain the MERA representation of the 1D cluster state ($|\psi_0\rangle$) such that it becomes a stable fixed point. When $|\psi_0\rangle$ is used as input, each convolution–pooling unit produces the same state $|\psi_0\rangle$ with reduced system size in the unmeasured qubits, while yielding deterministic outcomes ($X=1$) in the measured qubits. The fully connected layer measures the SOP for $|\psi_0\rangle$. When an input wavefunction is perturbed away from $|\psi_0\rangle$, our QCNN corrects such ‘errors’. For example, if a single X error occurs, the first pooling layer identifies its location, and controlled unitary operations correct the error propagated through the circuit (Fig. 3). Similarly, if an initial state has multiple, sufficiently separated errors (possibly in coherent superpositions), the error density after several iterations of convolution and pooling layers will be substantially smaller³⁵. If the input state converges to the fixed point, our QCNN classifies it into the SPT phase with high fidelity. Clearly, this mechanism resembles the classification of quantum phases based on renormalization-group flow.

Obtaining QCNN from training procedure. Having analytically illustrated the computational power of the QCNN circuit model, we now demonstrate how a QCNN for \mathcal{P} can also be obtained using the learning procedure. Details of the hyperparameters of the QCNN can be found in the Methods and Supplementary Fig. 2. Initially, all unitaries are set to random values. As classically simulating our training procedure requires expensive computational resources, we focus on a relatively small system with $N=15$ spins and QCNN depth $d=1$; there are a total of 1,309 parameters to be learned (see Methods). Our training data consists of 40 evenly spaced points along the line $h_2=0$, where the Hamiltonian is exactly solvable by the Jordan–Wigner transformation. Using gradient descent with the MSE function (1), we iteratively update the unitaries until convergence (see Methods). The classification output of the resulting QCNN for generic h_2 is shown in Fig. 4. This QCNN accurately reproduces the two-dimensional phase diagram over the entire parameter regime, despite being trained only on samples from a set of solvable points that do not even cross the lower phase boundary.

This example illustrates how the QCNN structure avoids overfitting to training data with its exponentially reduced number of parameters. While the training dataset for this particular QPR prob-

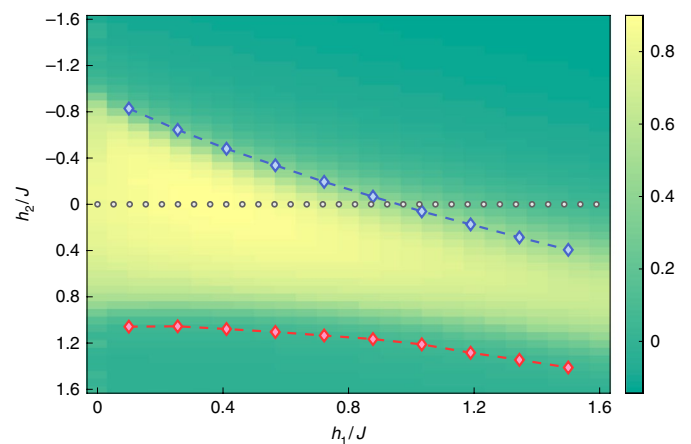


Fig. 4 | Output of a trained QCNN. We numerically optimize our QCNN for a system of $N=15$ spins and $d=1$ starting from random initial values. The training data points are 40 equally spaced points $h_1 \in [0, 2]$ along the line $h_2=0$ where the Hamiltonian is solvable by the Jordan–Wigner transformation (grey dots). The blue and red diamonds are phase boundary points extracted from infinite-size DMRG numerical simulations, while the background shading (colour scale) represents the expectation value of QCNN output.

lem consists of solvable points, more generally, such a dataset can be obtained by using traditional methods (such as measuring SOPs) to classify representative states that can be efficiently generated either numerically or experimentally^{36,37}.

Optimizing QEC

As seen in the previous example, the QCNN’s architecture enables one to perform effective QEC. We next leverage this feature to design a new QEC code that is itself optimized for a given error model. More specifically, any QCNN circuit (and its inverse) can be viewed as a decoding (encoding) quantum channel between the physical input qubits and the logical output qubit. The encoding scheme introduces sets of new qubits in a predetermined state, for example $|0\rangle$, while the decoding scheme performs measurements (Fig. 5a). Given an error channel \mathcal{N} , our aim is therefore to maximize the recovery fidelity

$$f = \sum_{|\psi_l\rangle \in \{\pm x, y, z\}} \langle \psi_l | \mathcal{M}^{-1}(\mathcal{N}(\mathcal{M}(|\psi_l\rangle \langle \psi_l|))) | \psi_l \rangle \quad (5)$$

where $\mathcal{M}(\mathcal{M}^{-1})$ is the encoding (decoding) scheme generated by a QCNN circuit, and $|\pm x, y, z\rangle$ are the ± 1 eigenstates of the Pauli matrices. Thus, our method simultaneously optimizes both encoding and decoding schemes, while ensuring their efficient implementation in realistic systems. The variational optimization can be carried out with an unknown \mathcal{N} , since f can be evaluated experimentally.

To illustrate the potential of this procedure, we consider a two-layer QCNN with $N=9$ physical qubits and 126 variational parameters (Fig. 5a and Methods). This particular architecture includes the nested (classical) repetition codes and the 9-qubit Shor code³⁸; in the following, we compare our performance to the better of the two. We consider three different input error models: (1) independent single-qubit errors on all qubits with equal probabilities p_μ for $\mu=X, Y$ and Z errors or (2) anisotropic probabilities $p_x \neq p_y = p_z$, and (3) independent single-qubit anisotropic errors with additional two-qubit correlated errors $X_i X_{i+1}$ with probability p_{xx} .

On initializing all QCNN parameters to random values and numerically optimizing them to maximize f , we find that our model

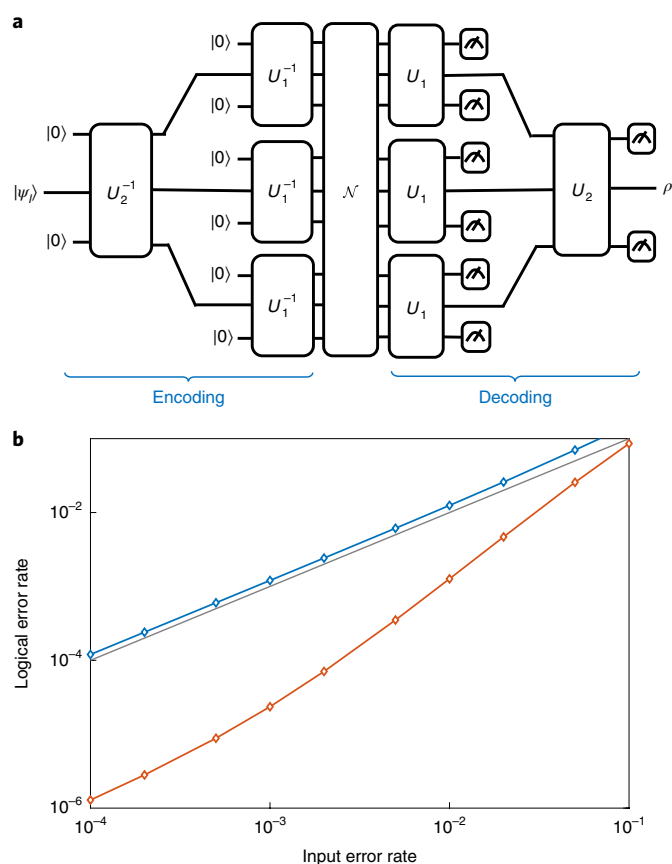


Fig. 5 | QCNN for optimizing quantum error correction. **a**, Schematic for using QCNNs to optimize QEC. The inverse QCNN encodes a single logical qubit $|\psi_L\rangle$ into nine physical qubits, which undergo noise \mathcal{N} . The QCNN then decodes these to obtain the logical state ρ . Our aim is to maximize $\langle \psi_L | \rho | \psi_L \rangle$. **b**, Logical error rate of Shor code (blue) versus a learned QEC code (orange) in a correlated error model. The input error rate is defined as the sum of all probabilities p_i and $p_{i,i'}$. The performance of the Shor code is worse than performing no error correction at all (identity, grey line), while the optimized code can still substantially reduce the error rate.

produces the same logical error rate as known codes in case (1) but can reduce the error rate by a constant factor of up to 50% in case (2), depending on the specific input error probability ratios (see Methods and Supplementary Fig. 4). In case (3), the optimized QEC code performs substantially better than known codes (Fig. 5b). As the Shor code is only guaranteed to correct arbitrary single-qubit errors, it performs more poorly than using no error correction, while the optimized QEC code performs much better. This example demonstrates the power of using QCNNs to obtain and optimize new QEC codes for realistic, a priori unknown error models.

Experimental realizations

Our QCNN architecture can be efficiently implemented on several state-of-the-art experimental platforms. The key ingredients for realizing QCNNs include the efficient preparation of quantum many-body input states, the application of two-qubit gates at various length scales and projective measurements. As in stabilizer-based QEC, the measurements of intermediate qubits and feed-forwarding can be replaced by controlled two-qubit unitary operations so that measurements are performed only at the end of an experimental sequence. These capabilities have already been demonstrated in multiple programmable quantum simulators consisting of $N \geq 50$ qubits based on trapped neutral atoms and ions, or superconducting qubits^{39–42}.

As an example, we present a feasible protocol for near-term implementation of our exact cluster model QCNN circuit via neutral Rydberg atoms^{39,43}, where long-range dipolar interactions allow high-fidelity entangling gates⁴⁴ among distant qubits in a variable geometric arrangement. The qubits can be encoded in the hyperfine ground states, where one of the states can be coupled to the Rydberg level to perform efficient entangling operations via the Rydberg-blockade mechanism⁴⁴; an explicit implementation scheme for every gate in Fig. 2b is provided in the Methods. Our QCNN at depth d with N input qubits requires at most $\frac{2N}{2}(1 - 3^{1-d}) + N3^{1-d}$ multi-qubit operations and $4d$ single-qubit rotations. For a realistic effective coupling strength $\Omega \approx 2\pi \times 10\text{--}100$ MHz and single-qubit coherence time $\tau \approx 200\text{ }\mu\text{s}$ limited by the Rydberg state lifetime, approximately $\Omega\tau \approx 2\pi \times 10^3\text{--}10^4$ multi-qubit operations can be performed, and a $d=4$ QCNN on $N \approx 100$ qubits is feasible. These estimates are reasonably conservative as we have not considered advanced control techniques such as pulse-shaping⁴⁵ or potentially parallelizing independent multi-qubit operations.

Outlook

These considerations indicate that QCNNs provide a promising quantum machine learning paradigm. Several interesting generalizations and future directions can be considered. For example, while we have only presented the QCNN circuit structure for recognizing 1D phases, it is straightforward to generalize the model to higher dimensions, where phases with intrinsic topological order such as the toric code are supported^{46,47,48}. Such studies could potentially identify nonlocal order parameters with low sample complexity for lesser-understood phases such as quantum spin liquids⁴⁹ or anyonic chains⁵⁰. To recognize more exotic phases, we could also relax the translation-invariance constraints, resulting in $O(N \log(N))$ parameters for system size N , or use ancilla qubits to implement parallel feature maps following traditional CNN architecture. Further extensions could incorporate optimizations for fault-tolerant operations on QEC code spaces. Finally, while we have used a finite-difference scheme to compute gradients in our learning demonstrations, the structural similarity of a QCNN to its classical counterpart suggests the possibility of adopting more efficient schemes such as backpropagation¹.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, statements of code and data availability and associated accession codes are available at <https://doi.org/10.1038/s41567-019-0648-8>.

Received: 9 October 2018; Accepted: 23 July 2019;

Published online: 26 August 2019

References

1. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
2. Lin, H. W., Tegmark, M. & Rolnick, D. Why does deep and cheap learning work so well? *J. Stat. Phys.* **168**, 1223–1247 (2017).
3. Mehta, P. & Schwab, D. J. An exact mapping between the variational renormalization group and deep learning. Preprint at <https://arxiv.org/abs/1410.3831> (2014).
4. Carleo, G. & Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **355**, 602–606 (2017).
5. van Nieuwenburg, E. P. L., Liu, Y. H. & Huber, S. D. Learning phase transitions by confusion. *Nat. Phys.* **13**, 435–439 (2017).
6. Carrasquilla, J. & Melko, R. G. Machine learning phases of matter. *Nat. Phys.* **13**, 431–434 (2017).
7. Wang, L. Discovering phase transitions with unsupervised learning. *Phys. Rev. B* **94**, 195105 (2016).
8. Levine, Y., Cohen, N. & Shashua, A. Quantum entanglement in deep learning architectures. *Phys. Rev. Lett.* **122**, 065301 (2019).
9. Zhang, Y. & Kim, E.-A. Quantum loop topography for machine learning. *Phys. Rev. Lett.* **118**, 216401 (2017).

10. Maskara, N., Kubica, A. & Jochym-O'Connor, T. Advantages of versatile neural-network decoding for topological codes. *Phys. Rev. A* **99**, 052351 (2019).
11. Haah, J., Harrow, A. W., Ji, Z., Wu, X. & Yu, N. Sample-optimal tomography of quantum states. *IEEE Trans. Inform. Theory* **63**, 5628–5641 (2017).
12. Lee, J. Y. & Landon-Cardinal, O. Practical variational tomography for critical one-dimensional systems. *Phys. Rev. A* **91**, 062128 (2019).
13. Ladd, T. D. et al. Quantum computers. *Nature* **464**, 45–53 (2010).
14. Monroe, C. & Kim, J. Scaling the ion trap quantum processor. *Science* **339**, 1164–1169 (2013).
15. Devoret, M. H. & Schoelkopf, R. J. Superconducting circuits for quantum information: an outlook. *Science* **339**, 1169–1174 (2013).
16. Awschalom, D. D., Bassett, L. C., Dzurak, A. S., Hu, E. L. & Petta, J. R. Quantum spintronics: engineering and manipulating atom-like spins in semiconductors. *Science* **339**, 1174–1179 (2013).
17. Biamonte, J. et al. Quantum machine learning. *Nature* **549**, 195–202 (2017).
18. Dunjko, V., Taylor, J. M. & Briegel, H. J. Quantum-enhanced machine learning. *Phys. Rev. Lett.* **117**, 130501 (2016).
19. Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. Preprint at <https://arxiv.org/abs/1802.06002> (2018).
20. Huggins, W., Patil, P., Mitchell, B., Whaley, K. B. & Stoudenmire, E. M. Towards quantum machine learning with tensor networks. *Quantum Sci. Technol.* **4**, 024001 (2018).
21. Huang, C.-Y., Chen, X. & Lin, F.-L. Symmetry-protected quantum state renormalization. *Phys. Rev. B* **88**, 205124 (2013).
22. Singh, S. & Vidal, G. Symmetry-protected entanglement renormalization. *Phys. Rev. B* **88**, 121108(R) (2013).
23. Kim, I. & Swingle, B. Robust entanglement renormalization on a noisy quantum computer. Preprint at <https://arxiv.org/abs/1711.07500> (2017).
24. LeCun, Y. & Bengio, Y. in *The Handbook of Brain Theory and Neural Networks* 255–258 (MIT Press, 1995).
25. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Proc. 25th Int. Conf. Neural Information Processing Systems* Vol. 1, 1097–1105 (2012).
26. Vidal, G. Class of many-body states that can be efficiently simulated. *Phys. Rev. Lett.* **101**, 110501 (2008).
27. Aguado, M. & Vidal, G. Entanglement renormalization and topological order. *Phys. Rev. Lett.* **100**, 070404 (2008).
28. Pfeifer, R. N. C., Evenbly, G. & Vidal, G. Entanglement renormalization, scale invariance, and quantum criticality. *Phys. Rev. A* **79**, 040301 (2009).
29. Preskill, J. *Lecture Notes for Physics 229: Quantum Information and Computation*. (California Institute of Technology, 1998).
30. Sachdev, S. *Quantum Phase Transitions* (Cambridge University Press, 2011).
31. Haldane, F. D. M. Nonlinear field theory of large-spin Heisenberg antiferromagnets: semiclassically quantized solitons of the one-dimensional easy-axis Néel state. *Phys. Rev. Lett.* **50**, 1153 (1983).
32. Haegeman, J., Pérez-García, D., Cirac, I. & Schuch, N. Order parameter for symmetry-protected topological phases in one dimension. *Phys. Rev. Lett.* **109**, 050402 (2012).
33. Pollmann, F. & Turner, A. M. Detection of symmetry-protected topological phases in one dimension. *Phys. Rev. B* **86**, 125441 (2012).
34. Brown, L. D., Cai, T. T. & DasGupta, A. Interval estimation for a binomial proportion. *Stat. Sci.* **16**, 101 (2001).
35. Zeng, B. & Zhou, D. L. Topological and error-correcting properties for symmetry-protected topological order. *Eur. Phys. Lett.* **113**, 56001 (2016).
36. Schwarz, M., Temme, K. & Verstraete, F. Preparing projected entangled pair states on a quantum computer. *Phys. Rev. Lett.* **108**, 110502 (2012).
37. Ge, Y., Molnar, A. & Cirac, J. I. Rapid adiabatic preparation of injective projected entangled pair states and Gibbs states. *Phys. Rev. Lett.* **116**, 080503 (2016).
38. Shor, P. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A* **52**, R2493(R) (1995).
39. Bernien, H. et al. Probing many-body dynamics on a 51-atom quantum simulator. *Nature* **551**, 579–584 (2017).
40. Zhang, J. et al. Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator. *Nature* **551**, 601–604 (2017).
41. Brydges, T. et al. Probing Rényi entanglement entropy via randomized measurements. *Science* **364**, 260–263 (2019).
42. Harris, R. et al. Phase transitions in a programmable quantum spin glass simulator. *Science* **361**, 162–165 (2018).
43. Labuhn, H. et al. Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum ising models. *Nature* **534**, 667–670 (2016).
44. Levine, H. et al. High-fidelity control and entanglement of Rydberg atom qubits. *Phys. Rev. Lett.* **121**, 123603 (2018).
45. Freeman, R. Shaped radiofrequency pulses in high resolution NMR. *Progr. Nucl. Magn. Reson. Spectrosc.* **32**, 59–106 (1998).
46. Kitaev, A. Y. Fault-tolerant quantum computation by anyons. *Ann. Phys.* **303**, 2–30 (2003).
47. Levin, M. A. & Wen, X.-G. String-net condensation: a physical mechanism for topological phases. *Phys. Rev. B* **71**, 045110 (2005).
48. Schuch, N., Pérez-García, D. & Cirac, J. I. PEPS as ground states: degeneracy and topology. *Ann. Phys.* **325**, 2153 (2010).
49. Savary, L. & Balents, L. Quantum spin liquids: a review. *Rep. Prog. Phys.* **80**, 016502 (2017).
50. Feiguin, A. et al. Interacting anyons in topological quantum liquids: the golden chain. *Phys. Rev. Lett.* **98**, 160409 (2007).

Acknowledgements

We thank I. Cirac, E. Farhi, W. W. Ho, C. Nayak, H. Pichler, J. Preskill, X. Qi, A. Vishwanath, Z. Wang and X.-G. Wen for insightful discussions. I.C. acknowledges support from the Paul and Daisy Soros Fellowship, the Alfred Spector and Rhonda Kost Fellowship of the Hertz Foundation, and the Department of Defense through the National Defense Science and Engineering Graduate Fellowship Program. S.C. acknowledges support from the Miller Institute for Basic Research in Science. This work was supported through the National Science Foundation, the Center for Ultracold Atoms, the Vannevar Bush Faculty Fellowship and Google Research Award.

Author contributions

All authors contributed extensively to the work presented in this paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at <https://doi.org/10.1038/s41567-019-0648-8>.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to S.C.

Peer review information: *Nature Physics* thanks Zohar Ringel and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature Limited 2019

Methods

Phase diagram and QCNN circuit simulations. The phase diagram in Fig. 2a was numerically obtained using the infinite-size density matrix renormalization group (DMRG) algorithm. We generally follow the method outlined in ref.⁵¹ with 150 as the maximum bond dimension. To extract each data point in Fig. 2a, we numerically obtain the ground state energy density as a function of h_2 for fixed h_1 and compute its second-order derivative. The phase boundary points are identified from sharp peaks.

The simulation of our QCNN in Fig. 2b also uses matrix product state representations. We first obtain the input ground state wavefunction using finite-size DMRG⁵¹ with bond dimension $D=130$ for a system of $N=135$ qubits. We then perform the circuit operations by sequentially applying swap gates and two-qubit gates on the nearest neighboring qubits⁵². Each three-qubit gate is decomposed into two-qubit unitaries⁵³. We find that increasing the bond dimension to $D=150$ does not lead to any visible changes in our main figures, confirming a reasonable convergence of our method. The colour plot in Fig. 2a is similarly generated for a system of $N=45$ qubits.

QCNN for the $S=1$ Haldane chain. As discussed in the main text, the (spin-1/2) 1D cluster state belongs to an SPT phase protected by $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry, a phase that also contains the celebrated $S=1$ Haldane chain³¹. It is thus natural to ask whether this circuit can be used to detect the phase transition between the Haldane phase and an $S=1$ paramagnetic phase, which we numerically demonstrate here.

The one-parameter family of Hamiltonians we consider for the Haldane phase is defined on a 1D chain of N spin-1 particles with open boundary conditions³¹:

$$H_{\text{Haldane}} = J \sum_{j=1}^N \mathbf{S}_j \cdot \mathbf{S}_{j+1} + \omega \sum_{j=1}^N (\mathbf{S}_j^x)^2 \quad (6)$$

In this equation, \mathbf{S}_j denotes the vector of $S=1$ spin operators at site j , and J, ω are parameters of the Hamiltonian. The system is protected by a $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry generated by global π -rotations of every spin around the X and Y axes: $R_x = \prod_j e^{i\pi S_j^x}$, $R_y = \prod_j e^{i\pi S_j^y}$. When ω is zero or small compared to J , the ground state belongs to the SPT phase, but when ω/J is sufficiently large, the ground state becomes paramagnetic³¹.

To apply our QCNN circuit to this Haldane phase, we must first identify a quasilocal isometric map U between the two models, because their representations of the symmetry group are distinct. More specifically, since the cluster model has a $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry generated by $X_{\text{even(odd)}} = \prod_{i \text{ even(odd)}} X_i$, we require $UR_x U^\dagger = X_{\text{odd}}$

and $UR_y U^\dagger = X_{\text{even}}$. Such a map can be constructed following ref.⁵⁴. Intuitively, it extends the local Hilbert space of a spin-1 particle by introducing a spin singlet state $|s\rangle$ and mapping it to a pair of spin-1/2 particles: $|x\rangle \mapsto |+-\rangle$, $|y\rangle \mapsto -|+-\rangle$, $|z\rangle \mapsto -|-+\rangle$, $|s\rangle \mapsto |++\rangle$. Here, $|\pm\rangle$ denote the $|\pm 1\rangle$ eigenstates of the (spin-1/2) Pauli matrix X . $|\mu\rangle$ denotes a spin-1 state defined by $R_\nu |\mu\rangle = (-1)^{\delta_{\nu,\mu}+1} |\mu\rangle$, $\nu \in \{x, y, z\}$.

The QCNN circuit for the Haldane chain thus consists of applying U followed by the circuit presented in the main text.

Supplementary Fig. 1 shows the QCNN output for an input system of $N=54$ spin-1 particles at $d=1, \dots, 4$, obtained using matrix product state simulations with $D=160$. For this system size, we numerically identified the critical point as $\omega/J = 1.035 \pm 0.005$ by using DMRG to obtain the second derivative of energy density as a function of ω and J . The QCNN provides accurate identification of the phase transition.

Multiscale string order parameters. We examine the final operator measured by our circuit that recognizes the SPT phase in the Heisenberg picture. Although a QCNN performs non-unitary measurements in the pooling layers, similar to QEC circuits²⁹, one can postpone all measurements to the end and replace pooling layers by unitary-controlled gates acting on both measured and unmeasured qubits. In this way, the QCNN is equivalent to measuring a non-local observable

$$\mathcal{O} = (U_{\text{CP}}^{(d)} \dots U_{\text{CP}}^{(1)})^\dagger Z_{i-1} X_i Z_{i+1} (U_{\text{CP}}^{(d)} \dots U_{\text{CP}}^{(1)}) \quad (7)$$

where i is the index of the measured qubit in the final layer and $U_{\text{CP}}^{(l)}$ is the unitary corresponding to the convolution–pooling unit at depth l . A more explicit expression of \mathcal{O} can be obtained by commuting U_{CP} with the Pauli operators, which yields recursive relations:

$$U_{\text{CP}}^\dagger X_i U_{\text{CP}} = X_{i-2} X_i X_{i+2} \quad (8)$$

$$U_{\text{CP}}^\dagger Z_i U_{\text{CP}} = \frac{1}{2} (Z_i + Z_{i-2} X_{i-1} + X_{i+1} Z_{i+2} - Z_{i-2} X_{i-1} Z_i X_{i+1} Z_{i+2}) \quad (9)$$

\bar{i} enumerates every qubit at depth $l-1$, including those measured in the pooling layer. It follows that an SOP of the form $ZXX \dots XZ$ at depth l transforms into a weighted linear combination of 16 products of SOPs at depth $l-1$. Thus, instead

of measuring a single SOP, our QCNN circuit measures a sum of products of exponentially many different SOPs:

$$\mathcal{O} = \sum_{ab} C_{ab}^{(1)} S_{ab} + \sum_{a_1 b_1 a_2 b_2} C_{a_1 b_1 a_2 b_2}^{(2)} S_{a_1 b_1} S_{a_2 b_2} + \dots \quad (10)$$

\mathcal{O} can be viewed as a multiscale SOP with coefficients computed recursively in d using equations (8) and (9). This allows the QCNN to produce a sharp classification output even when the correlation length is as long as 3^d .

Demonstration of learning procedure for QPR. To perform our learning procedure in a QPR problem, we choose the hyperparameters for the QCNN as shown in Supplementary Fig. 2. This hyperparameter structure can be used for generic (1D) phases, and is characterized by a single integer n that determines the reduction of system size in each convolution–pooling layer, $L \rightarrow L/n$. (Supplementary Fig. 2 shows the special case where $n=3$.) The first convolution layer involves $(n+1)$ -qubit unitaries starting on every n th qubit. This is followed by n layers of n -qubit unitaries arranged as shown in Supplementary Fig. 2. The pooling layer measures $n-1$ out of every contiguous block of n qubits; each of these is associated with a unitary V_j applied to the remaining qubit, depending on the measurement outcome. This set of convolution and pooling layers is repeated d times. Finally, the fully connected layer consists of an arbitrary unitary on the remaining N/n^d qubits, and the classification output is given by the measurement output of the middle qubit (or any fixed choice of one of them). For our example, we choose $n=3$ because the Hamiltonian in equation (2) involves three-qubit terms.

In our simulations, we consider only $N=15$ spins and $d=1$, because simulating quantum circuits on classical computers requires a large amount of resources. We parameterize unitaries as exponentials of generalized $\alpha \times \alpha$ Gell-Mann matrices $\{\Lambda_i\}$, where $\alpha=2^v$, and v is the number of qubits involved in the unitary⁵⁵:

$$U = \exp \left(-i \sum_j c_j \Lambda_j \right).$$

This parameterization is used directly for the unitaries in the convolution layers $C_2 \dots C_n$, the pooling layer and the fully connected layer. For the first convolution layer C_1 , we restrict the choice of U_1 to a product of six two-qubit unitaries between each possible pair of qubits: $U_1 = U_{(23)} U_{(24)} U_{(13)} U_{(14)} U_{(12)} U_{(34)}$, where $U_{(\alpha\beta)}$ is a two-qubit unitary acting on qubits indexed by α and β . Such a decomposition is useful when considering experimental implementation.

In the QCNN learning procedure, all parameters c_μ are set to random values between 0 and 2π for the unitaries $\{U, V, F\}$. In every iteration of gradient descent, we compute the derivative of the mean-squared error function (equation (1)) to first order with respect to each c_μ by using the finite-difference method:

$$\frac{\partial \text{MSE}}{\partial c_\mu} = \frac{1}{2\epsilon} (\text{MSE}(c_\mu + \epsilon) - \text{MSE}(c_\mu - \epsilon)) + O(\epsilon^2) \quad (11)$$

Each coefficient is thus updated as $c_\mu \mapsto c_\mu - \eta \frac{\partial \text{MSE}}{\partial c_\mu}$, where η is the learning rate for that iteration. We compute the learning rate using the bold driver technique from machine learning, where η is increased by 5% if the error has decreased from the previous iteration and decreased by 50% otherwise⁵⁶. We repeat the gradient descent procedure until the error function changes on the order of 10^{-5} between successive iterations. In our simulations, we use $\epsilon = 10^{-4}$ for the gradient computation and begin with an initial learning rate of $\eta_0 = 10$.

Construction of the QCNN circuit. To construct the exact QCNN circuit in Fig. 2b, we follow the guidelines discussed in the main text. Specifically, we design the convolution and pooling layers to satisfy the following two important properties:

1. Fixed-point criterion. If the input is a cluster state $|\psi_0\rangle$ of L spins, the output of the convolution–pooling layers is a cluster state $|\psi_0\rangle$ of $L/3$ spins, with all measurements deterministically yielding $|0\rangle$.
2. QEC criterion. If the input is not $|\psi_0\rangle$ but instead differs from $|\psi_0\rangle$ at one site by an error that commutes with the global symmetry, the output should still be a cluster state of $L/3$ spins, but at least one of the measurements will result in the state $|1\rangle$.

These two properties are desirable for any quantum circuit implementation of RG flow for performing QPR.

In the specific case of our Hamiltonian, the ground state (1D cluster state) is a graph state, which can be efficiently obtained by applying a sequence of controlled-phase gates to a product state. This simplifies the construction of the MERA representation for the fixed-point criterion. To satisfy the QEC criterion, we treat the ground state manifold of the unperturbed Hamiltonian $H = -J \sum_i Z_i X_{i+1} Z_{i+2}$ as the code space of a stabilizer code with stabilizers $\{Z_i X_{i+1} Z_{i+2}\}$. The remaining degrees of freedom in the QCNN convolution and pooling layers are then specified such that the circuit detects and corrects the error (that is, it measures at least one $|1\rangle$ and prevents propagation to the next layer) when a single-qubit X error is present.

QCNN for general QPR problems. Our interpretation of QCNNs in terms of MERA and QEC motivates their application for recognizing more generic quantum phases. For any quantum phase \mathcal{P} whose RG fixed-point wavefunction

$|\psi_0(\mathcal{P})\rangle$ has a tensor network representation in isometric or G -isometric form⁵⁷ (Supplementary Fig. 3a), one can systematically construct a corresponding QCNN circuit. This family of quantum phases includes all 1D SPT and 2D string-net phases^{47,57,58}. In these cases, one can explicitly construct a commuting parent Hamiltonian for $|\psi_0(\mathcal{P})\rangle$ and a MERA structure in which $|\psi_0(\mathcal{P})\rangle$ is a fixed-point wavefunction (Supplementary Fig. 3a for 1D systems). The diagrammatic proof of this fixed-point property is given in Supplementary Fig. 3b. Furthermore, any ‘local error’ perturbing an input state away from $|\psi_0(\mathcal{P})\rangle$ can be identified by measuring a fraction of terms in the parent Hamiltonian, similar to syndrome measurements in stabilizer-based QEC²⁹. Then, a QCNN for \mathcal{P} simply consists of the MERA for $|\psi_0(\mathcal{P})\rangle$ and a nested QEC scheme in which an input state with error density below the QEC threshold⁵⁹ ‘flows’ to the RG fixed point. Such a QCNN can be optimized via our learning procedure.

While our generic learning protocol begins with completely random unitaries, as in the classical case⁴, this initialization may not be the most efficient for gradient descent. Instead, motivated by deep learning techniques such as pre-training⁴, a better initial parameterization would consist of a MERA representation of $|\psi_0(\mathcal{P})\rangle$ and one choice of nested QEC. With such an initialization, the learning procedure serves to optimize the QEC scheme, expanding its threshold to the target phase boundary (Supplementary Fig. 3c).

Experimental resource analysis. To compute the gate depth of the cluster model QCNN circuit in a Rydberg atom implementation, we analyse each gate shown in Fig. 2b. By postponing pooling layer measurements to the end of the circuit, the multi-qubit gates required are

$$C_z Z_{ij} = e^{i\pi(-1+Z_i)(-1+Z_j)/4} \quad (12)$$

$$C_x Z_{ij} = e^{i\pi(-1+X_i)(-1+Z_j)/4} \quad (13)$$

$$C_x C_x X_{ijk} = e^{i\pi(-1+X_i)(-1+X_j)(-1+X_k)/8} \quad (14)$$

By using Rydberg blockade-mediated controlled gates⁶⁰, it is straightforward to implement $C_z Z_{ij}$ and $C_z Z_{ijk} = e^{i\pi(-1+Z_i)(-1+Z_j)(-1+Z_k)/8}$. The desired $C_x Z_{ij}$ and $C_x C_x X_{ijk}$ gates can then be obtained by conjugating $C_z Z_{ij}$ and $C_z Z_{ijk}$ by single-qubit rotations. For an input size of N spins, the k th convolution–pooling unit thus applies $4N/3^{k-1}$ $C_z Z_{ij}$ gates, $N/3^{k-1}$ $C_x C_x X_{ijk}$ gates and $2N/3^{k-1}$ layers of $C_x Z_{ij}$ gates. The depth of single-qubit rotations required is $4d$, as these rotations can be implemented in parallel on all N qubits. Finally, the fully connected layer consists of $N3^{1-d}$ $C_z Z_{ij}$ gates. Thus, the total number of multi-qubit operations required for a QCNN of depth d operating on N spins is $\frac{7N}{2}(1 - 3^{1-d}) + N3^{1-d}$. Note that we do not need to use SWAP gates since the Rydberg interaction is long-range.

Demonstration of learning procedure for QEC. To obtain the QEC code considered in the main text, we consider a QCNN with $N=9$ input physical qubits and simulate the circuit evolution of its $2^9 \times 2^9$ density matrix exactly. Strictly speaking, our QCNN has three layers: a three-qubit convolution layer U_1 , a 3-to-1 pooling layer and a 3-to-1 fully connected layer U_2 . Without loss of generality, we may ignore the optimization over the pooling layer by absorbing its effect into the first convolution layer, leading to the effective two-layer structure shown in Fig. 5a. The generic three-qubit unitary operations U_1 and U_2 are parameterized using 63 Gell-Mann coefficients each.

As discussed in the main text, we consider three different error models: (1) independent single-qubit errors on all qubits with equal probabilities p_μ for $\mu = X, Y$ and Z errors, (2) independent single-qubit errors on all qubits, with anisotropic probabilities $p_x \neq p_y = p_z$ and (3) independent single-qubit anisotropic errors with additional two-qubit correlated errors $X_i X_{i+1}$ with probability p_{xx} . More specifically, the first two error models are realized by applying a (generally anisotropic) depolarization quantum channel to each of the nine physical qubits:

$$\mathcal{N}_{1,i} : \rho \mapsto \left(1 - \sum_\mu p_\mu\right) \rho + \sum_\mu p_\mu \sigma_i^\mu \rho \sigma_i^\mu \quad (15)$$

with Pauli matrices σ_i^μ for $i \in \{1, 2, \dots, 9\}$ (the qubit indices are defined from bottom to top in Fig. 5a). For the anisotropic case, we trained the QCNN on various different error models with the same total error probability $p_x + p_y + p_z = 0.001$ but different relative ratios; the resulting ratio between the logical error probability of the Shor code and that of the QCNN code is plotted as a function of anisotropy in Supplementary Fig. 4. For strongly anisotropic models, the QCNN outperforms the Shor code, while for nearly isotropic models, the Shor code is optimal and QCNN can achieve the same logical error rate.

For the correlated error model, we additionally apply a quantum channel:

$$\mathcal{N}_{2,i} : \rho \mapsto (1 - p_{xx})\rho + p_{xx} X_i X_{i+1} \rho X_i X_{i+1} \quad (16)$$

for pairs of nearby qubits, that is $i \in \{1, 2, 4, 5, 7, 8\}$. Such a geometrically local correlation is motivated from experimental considerations. In this case, we train our QCNN circuit on a specific error model with parameter choices $p_z = 5.8 \times 10^{-3}$, $p_y = p_z = 2 \times 10^{-3}$, $p_x = 2 \times 10^{-4}$ and evaluate the logical error probabilities for various physical error models with the same relative ratios but different total error per qubit $p_x + p_y + p_z + p_{xx}$. In general, for an anisotropic logical error model with probabilities p_μ for σ_μ logical errors, the overlap f is $(1 - 2 \sum p_\mu/3)$, since $\langle \pm \nu | \sigma_\mu | \pm \nu \rangle = (-1)^{\delta_{\mu,\nu}+1}$. Because of this, we compute the total logical error probability from f as $1.5(1 - f)$. Hence, our goal is to maximize the logical state overlap f defined in equation (5). If we naively apply the gradient descent method based on f directly to both U_1 and U_2 , we find that the optimization is easily trapped in a local optimum. Instead, we optimize two unitaries U_1 and U_2 sequentially, similar to the layer-by-layer optimization in backpropagation for conventional CNN¹.

A few remarks are in order. First, since U_1 is optimized prior to U_2 , one needs to devise an efficient cost function C_1 that is independent of U_2 . In particular, simply maximizing f with an assumption on U_2 , for example that it equals the identity, may not be ideal, since such choice does not capture a potential interplay between U_1 and U_2 . Second, because U_1 captures arbitrary single-qubit rotations, the definition of C_1 should be basis-independent. Finally, we note that the tree structure of our circuit allows one to view the first layer as an independent quantum channel:

$$\mathcal{M}_{U_1} : \rho \mapsto \text{tr}_a \left[U_1 \mathcal{N} \left(U_1^\dagger (|0\rangle\langle 0| \otimes \rho \otimes |0\rangle\langle 0|) U_1 \right) U_1^\dagger \right] \quad (17)$$

where $\text{tr}_a[\cdot]$ denotes tracing over the ancilla qubits that are measured in the intermediate step. From this perspective, \mathcal{M}_{U_1} describes an effective error model to be corrected by the second layer.

With these considerations, we optimize U_1 such that the effective error model \mathcal{M}_{U_1} becomes as classical as possible, that is \mathcal{M}_{U_1} is dominated by a ‘flip’ error along a certain axis with a strongly suppressed ‘phase’ error. Only then will the remnant, simpler errors be corrected by the second layer. More specifically, one may represent \mathcal{M}_{U_1} using a map $\mathcal{M}_{U_1} : \mathbf{r} \mapsto M\mathbf{r} + \mathbf{c}$, where $\mathbf{r} \in \mathbb{R}^3$ is the Bloch vector for a qubit state $\rho \equiv \frac{1}{2} \mathbf{1} + \mathbf{r} \cdot \boldsymbol{\sigma}$, where $\mathbf{1}$ is the identity operator and $\boldsymbol{\sigma} = (X, Y, Z)$ is the vector of Pauli matrices⁶¹. The singular values of the real matrix M encode the probabilities $p_1 \geq p_2 \geq p_3$ for three different types of errors. We choose our cost function for the first layer as $C_1 = p_1^2 + p_2 + p_3$, which is relatively more sensitive to p_2 and p_3 than p_1 and ensure that the resultant, optimized channel \mathcal{M}_{U_1} is dominated by one type of error (with probability p_1). We note that M can be efficiently evaluated from a quantum device without knowing \mathcal{N} , by performing quantum process tomography for a single logical qubit. Once U_1 is optimized, we use gradient descent to find an optimal U_2 to maximize f . As with QPR, gradients are computed via the finite-difference method, and the learning rate is determined by the bold driver technique⁴.

Data availability

The data that support the plots within this paper and other findings of this study are available from the corresponding author on reasonable request.

References

- McCulloch, I. P. Infinite size density matrix renormalization group, revisited. Preprint at <https://arxiv.org/abs/0804.2509> (2008).
- Vidal, G. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.* **91**, 147902 (2003).
- Nielsen, M. A. & Chuang, I. *Quantum Computation and Quantum Information* (Cambridge Univ. Press, 2000).
- Verresen, R., Moessner, R. & Pollman, F. One-dimensional symmetry-protected topological phases and their transitions. *Phys. Rev. B* **96**, 165124 (2017).
- Bertlmann, R. A. & Krammer, P. Bloch vectors for qudits. *J. Phys. A* **41**, 235303 (2008).
- Hinton, G. *Lecture notes for CSC2515: Introduction to machine learning* (Univ. Toronto, 2007).
- Schuch, N., Pérez-García, D. & Cirac, J. I. Classifying quantum phases using matrix product states and projected entangled pair states. *Phys. Rev. B* **84**, 165139 (2011).
- Chen, X., Gu, Z.-C. & Wen, X.-G. Classification of gapped symmetric phases in one-dimensional spin systems. *Phys. Rev. B* **83**, 035107 (2011).
- Aharonov, D. & Ben-Or, M. in *Proc. 29th Annu. ACM Symp. on the Theory of Computing* 176–188 (ACM, 1997).
- Saffman, M., Walker, T. & Molmer, K. Quantum information with Rydberg atoms. *Rev. Mod. Phys.* **82**, 2313 (2010).